REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	9/26/02	Final Report April 02 – Sep 02

4. TITLE AND SUBTITLE

SBIR Phase I Final Report: Misleading information detection through probabilistic decision tree classifiers

5. FUNDING NUMBERS

DAAH01-02-C-R146

6. AUTHOR(S)

Dr. Gordon Dakin, Sankar Virdhagriswaran

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Crystaliz, Inc.
9 Damon Mill Square, Suite H2
Concord, MA 01742

8. PERFORMING ORGANIZATION REPORT NUMBER

MISINF000Z

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

DARPA
Defense SBIR Program Office
3701 North Fairfax Ave
Arlington, VA 22203

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

DI-MGMT-80368
DI-MISC-80711

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT (see Section 5.3b of this solicitation)

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

20021010 117

13. ABSTRACT (Maximum 200 words)

This report details our work in detecting misleading information which has focused on the detection of creative accounting practices through the analysis of SEC filing data. We have adopted a decision-tree approach to detecting red-flag conditions associated with creative accounting practices. Decision trees provide a natural way to express expert knowledge of red-flag evidence, and the resulting classifications can be explained in human terms. Focusing on the diagnostic analysis of numeric data from balance sheets and income statements, we have tested and evaluated creative accounting detection rules suggested by Mulford & Comiskey to demonstrate the validity of our approach.

We developed a datamining application with a graphical user interface (GUI) to support the semi-automatic construction and induction of decision trees for classifying SEC filings as "positive" (red-flag) or "negative" instances of accounting fraud. The application permits various degrees of user involvement and/or automatic supervised learning of decision rules from training sets: (1) decision rules may be hand-crafted and used verbatim by the system; (2) top-level rules may be specified by the user, leaving the system to generate the remaining rules; (3) existing rules may be refined, via auto-adjustment of split-points (numeric thresholds), to achieve user-specified sensitivity and selectivity values. A cross-validation mechanism was implemented, to evaluate the effectiveness of auto-generated decision trees.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Decision trees, Mining, Probabilistic reasoning, Mining	20
	16. PRICE CODE

17. SECURITY CLASSIFICTION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unclassified

A Web Site Mis-Information Detection System

9/26/02

Sponsored by

Defense Advanced Research Projects Agency (DOD)
(Information Awareness Office)

ARPA Order K475/47

Issued by U.S. Army Aviation and Missile Command Under
Contract No. DAAH01-02-C-R

Disclaimer

"The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either express or implied, of the Defense Advanced Research Projects Agency of the U.S. Government.

Approved for public release; distribution unlimited

1. Abstract

Our work in detecting misleading information has focused on the detection of creative accounting practices through the analysis of SEC filing data. We have adopted a decision-tree approach to detecting red-flag conditions associated with creative accounting practices. Decision trees provide a natural way to express expert knowledge of red-flag evidence, and the resulting classifications can be explained in human terms. Focusing on the diagnostic analysis of numeric data from balance sheets and income statements, we have tested and evaluated creative accounting detection rules suggested by Mulford & Comiskey[1] to demonstrate the validity of our approach.

We developed a datamining application with a graphical user interface (GUI) to support the semi-automatic construction and induction of decision trees for classifying SEC filings as "positive" (red-flag) or "negative" instances of accounting fraud. The application permits various degrees of user involvement and/or automatic supervised learning of decision rules from training sets: (1) decision rules may be hand-crafted and used verbatim by the system; (2) top-level rules may be specified by the user, leaving the system to generate the remaining rules; (3) existing rules may be refined, via auto-adjustment of split-points (numeric thresholds), to achieve user-specified sensitivity and selectivity values. A cross-validation mechanism was implemented, to evaluate the effectiveness of auto-generated decision trees.

2. Approach

The project addresses the goal of detecting misleading information by classifying incoming information as misleading or not. The classification is performed using decision tree rules and probabilistic reasoning. These decision tree rules can be generated in three ways:

- Manually by experts
- Manually by experts and further refined using a learning system
- Automatically generated by a learning system

---

[1] Mulford, C. W., Comiskey, E. E., "The Financial Numbers Game: Detecting Creative Accounting Practices", John Wiley & Sons, 2002."

In Phase-I we concentrated on validating the viability of the decision tree approach. We implemented a learning/mining system to help create decision trees automatically and semi-automatically and applied the decision trees to the classification of SEC filings. We believe that we proved that our approach was successful.

In this final report, we present the deliverables of the Phase I SBIR project and the experiments performed to validate the approach. First, we present the data mining application GUI and how it can be used. Second, we present the generation of decision trees. Third, we present the experiments conducted to validate the approach. Fourth, we present the data extraction and management approach used. Finally, we present future directions of the project.

3. Datamining Application

3.1. Datamining GUI

Our goal in implementing the datamining GUI was to support the manual, semi-automatic, and automatic construction of decision trees for classifying SEC filings as "positive" or "negative" instances of creative accounting. Manual, interactive tree construction allows the user to incorporate expert knowledge of red-flag indicators in SEC filing data. Decision trees may be alternatively be constructed automatically, starting at any node, including the root node.

Prior to decision tree construction, the application user specifies the filing features of interest, plus any ad-hoc feature constraints, to define the filing data sample. Positive class (red-flag) instances are interactively specified, defining the minority class of misleading filings, based on historical cases of creative accounting examples. The GUI downloads required SEC filing records and SIC percentile summaries from the Genre server.

As shown in Figure 1, the GUI displays sample data entries in a scatterplot view, within the 2-dimensional space of any selected pair of numeric features. Positive class (red-flag) entries are colored red, and negative class entries are blue. By viewing the positive and negative filing entries with respect to alternative feature pairs, the user can discover the features that are the most appropriate for creating new decision rules to effectively discriminate between the positive and negative classes.

Numeric decision tree rules are created by opening the "Decision Tree Rule" dialog (see Figure 2a) and selecting a scatterplot axis (i.e., X or Y), then dragging the relevant crosshair to the desired split point position in the scatterplot. To facilitate selection of decision rule thresholds, the dialog displays a score that is the weighted sum of the current sensitivity (proportion of correctly-classified positives) and selectivity (proportion of correctly-classified negatives) achieved by the crosshair position. The user may also declare which halfspace, above or below the given threshold, is to be regarded as the "positive" halfspace, thereby indicating which tree node is intended to receive red-flag items. Positive tree nodes are colored pink in the tree view.

An SEC filing record submitted to the decision tree for classification begins at the root node and traverses a path determined by the decision rules of the successively visited nodes, until a leaf node is reached. In the "all-or-none" classification scheme, the leaf's positive (pink) or negative (yellow) label determines the filing's final classification. An alternative classification scheme based on *locally-weighted logistic regression* is described in a later section.

As shown in Figure 2b, the "Numeric Features" dialog allows the specification of feature constraints, to restrict the set of SEC filings downloaded from the Genre server. The *Rate Of Change* and *SIC Percentile* checkboxes further specify whether the feature's annual rate-of-change and/or SIC percentile values should be used for the scatter plot values and decision rule splitting features. This allows the user to compare filing metrics and their periodic changes against the filings of companies in the same industry, as is typically required by financial red-flag checklists.

Figure 1 shows an application session in which a 2-level decision tree has been generated. The "Scatter Plot Features" dialog allows the user to select the X and Y-axis scatter plot features. The current X-axis feature, *current-assets-turnover-ratio* (SIC percentile) was selected to be the root node's splitting feature, with a split-point of 55.5126, as specified by the vertical crosshair position. The Y-axis feature, *depreciation* (annual rate-of-change, SIC percentile) was chosen as the splitting feature for the second child of the root node, with a split-point of 41.6689, as specified by the horizontal crosshair position. The splitting feature and split-point associated with each nonterminal node is displayed as the node's name.

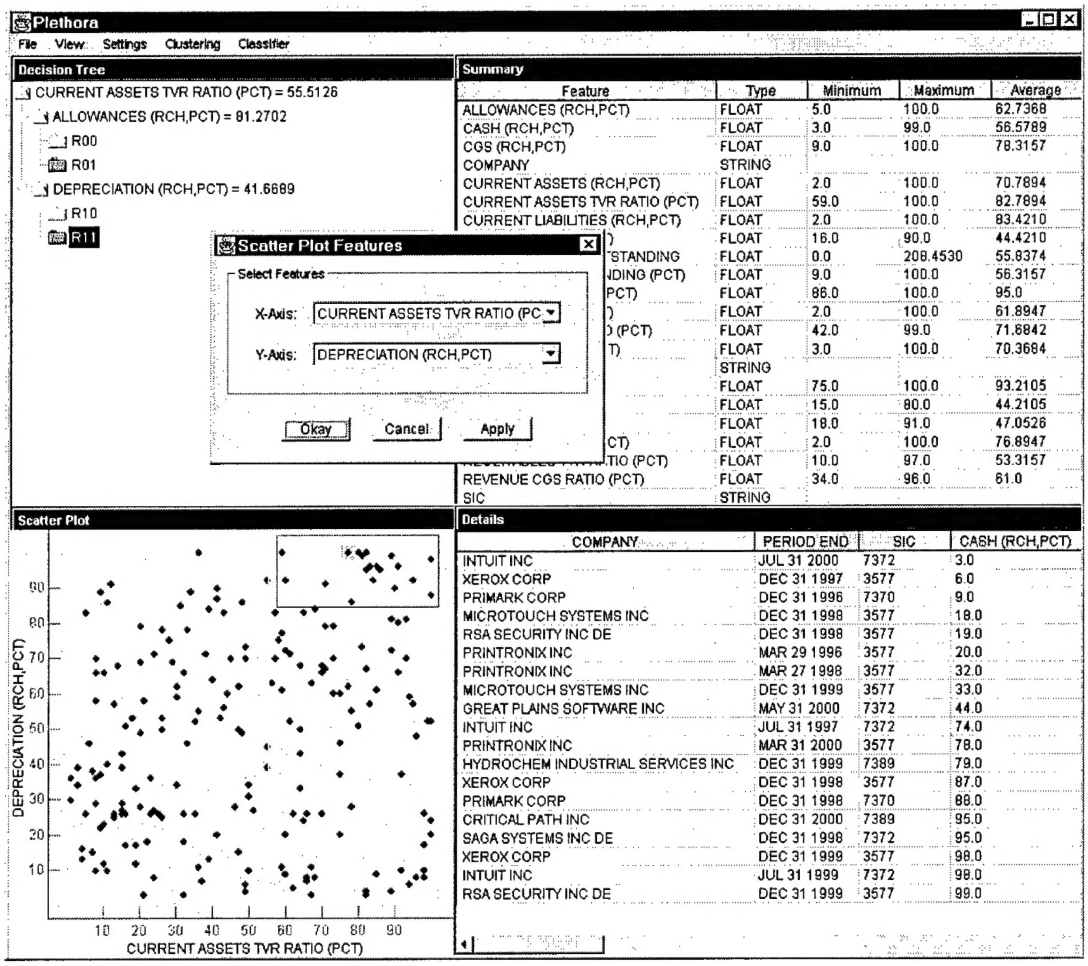


Figure 1. Datamining application.

To facilitate data exploration, the filing entries associated with the selected tree node are summarized in the "Summary" table, listed by row in the "Details" table, and highlighted in the scatterplot. The Summary table lists the minimum, maximum, and average value of each numeric feature. Scatterplot items may be selected with a rubber-rectangle, as seen in Figure 1, resulting in table displays of the selected items. Bi-directional linkage of selected items is supported between the scatterplot view and tables, whereby the item selected in a Details table row is surrounded by a blue square in the scatter plot.

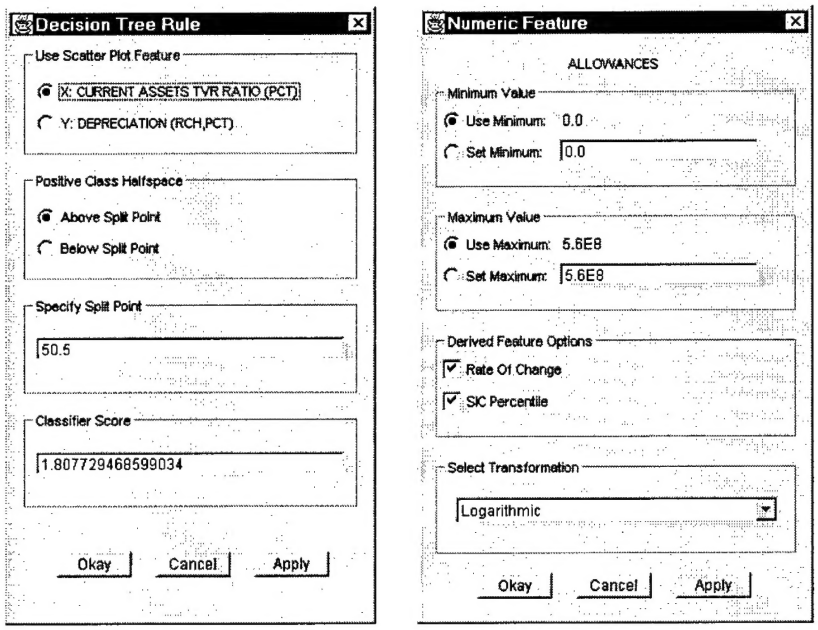


Figure 2. Decision rule and numeric feature dialogs.

3.2. Decision tree generation

In this section, we present two examples of decision trees that were constructed semi-automatically. Shown in Figure 3, roughly half of the decision rules were hand-crafted interactively via the "Decision Tree Rule" dialog, using the scatter plot display to visually compare the discriminability of alternative features, based on the distributions of historical "red-flag" SEC filing entries relative to negative filing entries. The remainder of the rules were generated automatically by selecting the "Generate Numeric Rule" menu item.

3.2.1. Feature selection

Mulford & Comisky's checklist for detecting premature or fictitious revenue includes the following criteria, among others: (1) unusual changes in revenue; (2) signs of overstated accounts receivable; and (3) questionable physical capacity to generate the reported revenue. The decision tree features of Figure 3 related to revenue changes are *total-revenues* (annual rate-of-change) and *revenue-CGS ratio*; the features related to accounts receivable are *receivables* (rate-of-change) and *days-sales-outstanding*; and the feature pertaining to physical capacity is *PP&E* (plant property & equipment) *turnover ratio.* All of these features are represented by their SIC percentiles, in order to contrast company filing metrics among industry peers.

The root decision rule for the first tree (a) in Figure 3 is based on the *revenue-CGS-ratio*. When this feature is paired with *total-revenues* in the scatter plot, it is apparent that more than half of the red-flag items occupy the upper halfspaces of the two features' percentile distributions. This observation motivated the creation of a decision rule for the *revenue-CGS-ratio* feature. Two additional decision rules were added to the tree, based on *days-sales-outstanding*, to further discriminate the items in the left side of the scatter plot, and *inventory-turnover-ratio*, to further discriminate the items in the upper right quadrant of the scatter plot.

In the second tree (b) of Figure 3, the decision rules for the root node and its first child node are based on the *receivables* and *CGS* (rate-of-change, percentile) features. As seen in the scatterplot, the chosen split points confine a large portion of negative items to the lower-left quadrant, while capturing a relatively high density of positive items in the upper right quadrant. Additional decision rules, including rules based on *PP&E-turnover-ratio*, *current-liabilities*, and *inventory-turnover-ratio*, were automatically generated, using a supervised learning mechanism described later.

3.2.2. Decision tree evaluation

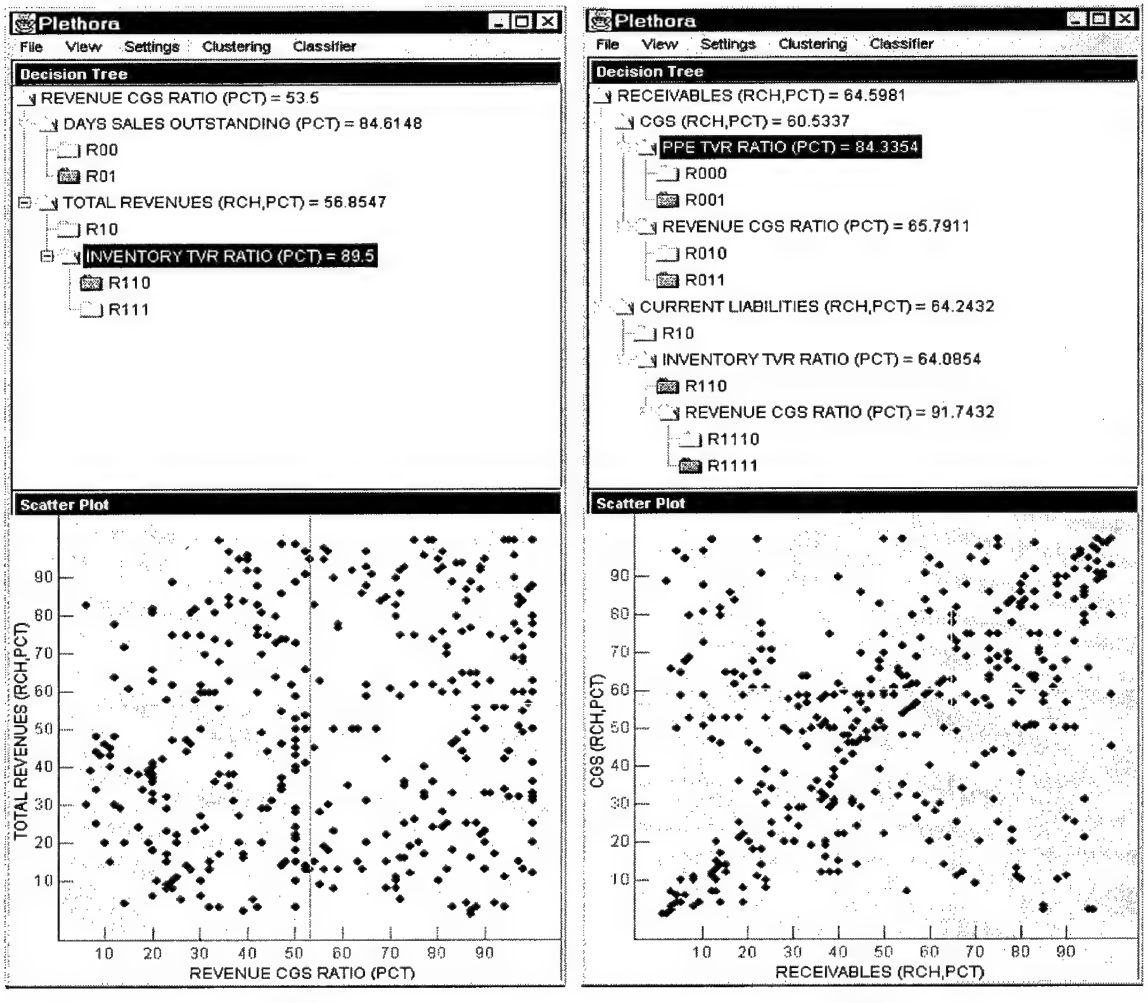
The decision trees shown in Figure 3 were generated semi-automatically with a "training set" of 394 10-K filings, including the 49 "positive class" examples listed in Appendix B. The 345 "negative" filings were obtained by sampling the filings of 10% of the companies within the 19 SICs spanned by the positive filing companies. To demonstrate the two trees' effectiveness in discriminating between the positive and negative training examples, classification error results are presented in Table 1, which shows the number of true positives (TP) , false positives (FP), true negatives (TN), and false negatives (FN). The table also includes each tree's sensitivity (TP / (TP + FN)), which measures the ability to classify training positives correctly; selectivity (TN / (TN + FP)), which measures the ability to classify training negatives correctly; and odds ratio ((TP · TN) / (FP · FN)), which is related to the odds of a correct classification.

The sensitivity of the decision trees can be increased at the expense of selectivity (or vice-versa) by adjusting the split-points used by the decision rules.

Tree	TP	FP	TN	FN	Sensitivity	Selectivity	Odds Ratio
a	36	77	268	13	0.73	0.78	9.64
b	34	78	267	15	0.69	0.77	7.76

Table 1. Decision tree classification results.

The apparent effectiveness of our semi-automatically generated decision trees in discriminating between positive and negative class instances is typical for a variety of trees constructed using various combinations of financial metrics. A more empirical evaluation of the numeric decision trees was performed using cross-validation, as described in a later section, to demonstrate the ability of automatically-generated trees to classify new (holdout) filing examples correctly.



(a) (b)

Figure 3. Decision tree examples.

3.2.3. Automatic generation of decision rules

The datamining application is capable of constructing decision trees automatically, either by incrementally adding splitting rules to selected leaf nodes, or by recursively growing the tree, starting from a selected tree leaf. The numeric rule generation procedure uses the selected node's filings (the set of filings that reach the node when classified by the tree) as a training set in a supervised learning paradigm. A decision tree can be built from scratch by invoking the tree generation routine for the root node, which is initially assigned the entire training set.

To generate a splitting rule for a given leaf node, the available numeric features are ranked according to a discriminability score, which in turn is based on the discriminability achieved by optimal placement of the split point. For a candidate splitting feature, the score for a split point position is calculated as a weighted sum of the sensitivity and selectivity achieved by the resulting split of the training set items associated with the node:

$$score = w * sensitivity + selectivity$$

The *sensitivity weight w* is a user-specified parameter. Other user-specified parameters state the minimum discriminability score allowed for splitting a node, and the maximum number of tree levels to generate at a time.

Once a new splitting rule has been auto-generated for a leaf node, it receives two child nodes, and the original leaf node's filing entries are split among the child nodes according to the splitting rule. In the "all-or-none" leaf-labeling scheme, each child node is labeled "positive" or "negative". By default, each child node is independently labeled positive if and only if it contains more positive entries than negative entries. If the user supplies an optional *sensitivity bias* parameter, then each child node is labeled positive if and only if its positive entry count, multiplied by the *sensitivity bias*, exceeds its negative entry count.

3.2.4. Locally-weighted logistic regression

In the "all-or-none" leaf-labeling scheme described above, an SEC filing is classified as positive if and only if its traversal through the decision tree ends at a "positive" tree node. This simple classification scheme is effective when the trainings sets are somewhat balanced. For very imbalanced training sets, in which the positive examples form a tiny minority, it is difficult to achieve a reasonable level of sensitivity without drastically increasing the

frequency of false positives. As an alternative to the all-or-none labeling of positive and negative nodes, a more flexible classification mechanism was implemented, based on *locally-weighted logistic regression* [2], to discriminate more effectively at the leaf level.

Global logistic regression produces a monotonic sigmoid function, which is used to predict the probability of an event occurring, given a set of independent input variables. In the SEC filing domain, we are concerned with predicting the probability that creative accounting has occurred, based on the multivariate metrics derived from the filings associated with a decision tree leaf. Logistic regression works well in the presence of noise, due to its smoothing characteristics. However, its monotonicity makes it insensitive to local clusters of positives or negatives.

In numeric feature space, red-flag SEC filing entries sometimes form two or more distinct clusters. Figure 4, for example, shows the scatter plot for a decision tree node's entries, viewed in the space of the features *net-income* and *current-liabilities* (showing SIC-percentiles for the features' rates-of-range). The red-flag filings form loose clusters in the lower-left corner, to the right of center, and (more loosely) near the upper-left corner. Each of these clusters is made up of several companies' filings.

Locally weighted logistic regression is a useful classification technique when local clusters, such as those in Figure 4, are present in the distribution of data. As a memory-based form of logistic regression, the sigmoid is calculated at the time of the query, using an inverse-distance formula to weight each training set entry according to its proximity to the query entry. This gives locally weighted logistic regression a nearest-neighbor behavior that respects the local distribution of data. However, locally weighted logistic regression has better smoothing characteristics than nearest-neighbor techniques.

A boolean application parameter indicates whether or not the decision trees should classify filings at the leaf level via locally-weighted logistic regression, in place of the "all-or-none" positive/negative leaf labeling scheme. A *logistic regression probability threshold* parameter also specifies the minimum probability of positive-class membership required for the decision tree to classify an input filing as a positive "red-flag" instance. Lowering the probability

[2] Deng, Kan. Omega: On-Line Memory-Based General-Purpose System Classifier. Ph.D. Dissertation, The Robotics Institute, School of Computer Science Carnegie Mellon University, 1998.

threshold effectively increases the decision tree's sensitivity, at the expense of selectivity. As shown in the cross-validation results of the next section, the locally weighted logistic regression approach appears to be preferable when the training set of positives and negatives is drastically imbalanced.

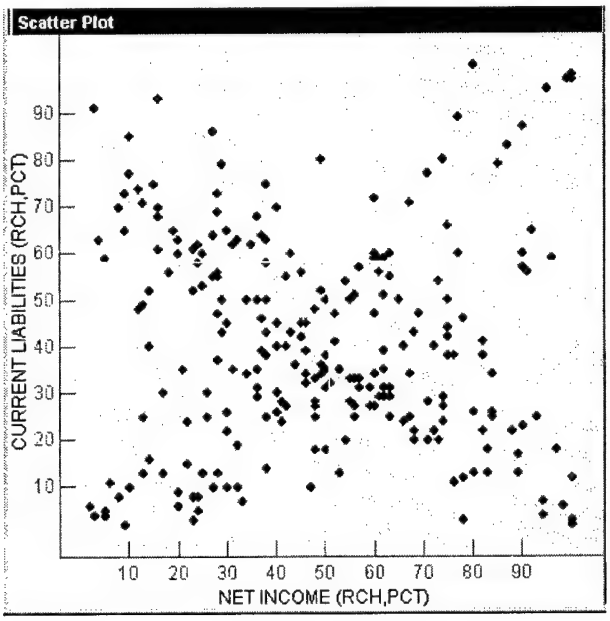


Figure 4. Clusters of red-flag filings.

3.2.5. Cross-validation

Cross-validation experiments were performed to test the effectiveness of automatically-generated decision trees, and to compare the degree of discriminability achieved by the "all-or-none" leaf labeling scheme versus the locally-weighted logistic regression approach. For each experiment, the experimental data set was randomly partitioned into N subsets, partitioning the positive and negative items separately. N combinations of $N - 1$ data subsets were formed by combining $N - 1$ training sets at a time, leaving N corresponding holdout sets for evaluating the decision trees generated with the training sets. Experimental results are shown for $N = 4$ and $N = 10$.

Three experimental data sets were used, each containing the 49 positive filing examples listed in Appendix B, together with different sets of "negative" filings. The 49 positive filings are from companies spanning 19 SICs, which together contain a total of 1647 companies. The

negative filings for the three experimental data sets were chosen by randomly sampling 5%, 10%, and 20% of those 1647 companies.

For each experimental data set, cross-validation was performed using both the "all-or-node" leaf labeling scheme, as well as the locally-weighted logistic regression scheme, for filing classification at the leaf-level. To produce varying sensitivity and selectivity behavior, the *sensitivity bias* parameter was varied when using the all-or-none leaf-labeling scheme, and the *logistic regression probability threshold* parameter was varied for the logistic regression scheme. As described previously, in the all-or-none leaf-labeling scheme, a newly-generated leaf node is labeled positive if and only if its positive entry count, multiplied by the *sensitivity bias*, exceeds its negative entry count.. For logistic regression, the *probability threshold* specifies the minimum probability of positive-class membership required for the decision tree to classify an input filing as a positive "red-flag" instance.

The tables below show the average sensitivity, selectivity, and odds ratio results for each cross-validation experiment. The comparative effectiveness of "all-or-none" leaf labeling versus logistic regression depends on the degree of imbalance between the positive and negative data sets. The all-or-none scheme is fairly effective for experiments whose negative class examples make up 5% and 10% of the 1647 companies, with the positive examples forming 0.21 and 0.12 of the experimental data sets, respectively. However, for the experiments involving 20% of the companies, with positives forming only 0.063 of the data sets, the all-or-none scheme fails to yield reasonable sensitivity levels, due to the scarcity of positives[3]. The locally-weighted logistic regression scheme yields comparatively better sensitivity, while retaining fairly high selectivity levels and odds ratios.

3.2.5.1. ALL-OR-NONE LEAF LABELING, WITH 5% OF THE COMPANIES (49 POSITIVES, 234 NEGATIVES)

N	Sensitivity Bias	Sensitivity	Selectivity	Odds Ratio
4	0.5	0.1250	0.9184	1.5813
4	1.0	0.3060	0.8702	2.9595
4	1.5	0.3060	0.8702	2.9595
10	0.5	0.1399	0.9628	4.2380
10	1.0	0.3800	0.8868	4.9460

[3] Experimental results in which no false positives occur produce an infinite odds ratio, which fails to reflect the accompanying lack of true positives.

10	1.5	0.4599	0.8540	5.1766

3.2.5.2. LOGISTIC REGRESSION, WITH 5% OF THE COMPANIES (49 POSITIVES, 234 NEGATIVES)

N	Probability Threshold	Sensitivity	Selectivity	Odds Ratio
4	0.2	0.5464	0.7402	3.5028
4	0.5	0.4455	0.8373	4.2098
4	0.8	0.3253	0.9187	5.4949
10	0.2	0.5850	0.7295	3.9150
10	0.5	0.3650	0.8160	2.5787
10	0.8	0.3050	0.9078	4.3598

3.2.5.3. ALL-OR-NONE LEAF LABELING, WITH 10% OF THE COMPANIES (49 POSITIVES, 394 NEGATIVES)

N	Sensitivity Bias	Sensitivity	Selectivity	Odds Ratio
4	0.5	0.1041	0.9709	3.8068
4	1.0	0.2660	0.9073	3.5321
4	1.5	0.2660	0.9073	3.5321
10	0.5	0.0800	0.9914	10.1333
10	1.0	0.1600	0.9594	4.6132
10	1.5	0.1600	0.9594	4.6132

3.2.5.4. LOGISTIC REGRESSION, WITH 10% OF THE COMPANIES (49 POSITIVES, 394 NEGATIVES)

N	Probability Threshold	Sensitivity	Selectivity	Odds Ratio
4	0.2	0.3269	0.8346	2.4497
4	0.5	0.3060	0.8840	3.3639
4	0.8	0.2051	0.9274	3.2820
10	0.2	0.2850	0.8492	2.2538
10	0.5	0.2050	0.8956	2.2008
10	0.8	0.0800	0.9247	1.0905

3.2.5.5. ALL-OR-NONE LEAF LABELING, WITH 20% OF THE COMPANIES (49 POSITIVES, 783 NEGATIVES)

N	Sensitivity Bias	Sensitivity	Selectivity	Odds Ratio
4	0.5	0.0000	1.0000	∞
4	1.0	0.0208	0.9781	0.9348
4	1.5	0.0993	0.9523	2.2694
10	0.5	0.0000	1.0000	∞
10	1.0	0.0850	0.9727	3.1733
10	1.5	0.0850	0.9727	3.1733

3.2.5.6. LOGISTIC REGRESSION, WITH 20% OF THE COMPANIES (49 POSITIVES, 783 NEGATIVES)

N	Probability Threshold	Sensitivity	Selectivity	Odds Ratio

4	0.2	0.3461	0.8896	4.2827
4	0.5	0.2868	0.9359	5.8468
4	0.8	0.2660	0.9604	8.7787
10	0.2	0.2400	0.9073	3.1764
10	0.5	0.1800	0.9414	3.6156
10	0.8	0.1200	0.9577	3.1642

3.2.6. Multiple decision tree deployment

In addition to implementing a datamining application to support the semi-automatic generation decision trees and demonstrating their effectiveness in classifying "red-flag" SEC filings, we also implemented a framework for parallel instantiation of multiple decision trees, as nodes a data-flow network. The parallel decision tree deployment capability exploits a data flow server, called JADCEA, which we developed for an earlier DARPA project. In the parallel deployment framework, *classifier nodes* are instantiated and "wired together" in a distributed dataflow network, which will support future experiments involving multiple, parallel classifiers.

In Phase II, we intend to investigate techniques for combining the results of multiple classifiers, including *stacking*, *cascading*, and *triaging* methods. Stacking approaches, including *Meta Decision Trees* [4], will be explored for combining the results of multiple existing decision trees, using supervised learning algorithms. Cascading techniques, in which multiple classifiers are applied in sequence, are especially appropriate for the imbalanced trainings sets of positives and negatives in our financial red-flag domain. In the cascading paradigm, the computationally simpler classifiers that are very highly sensitive and only moderately selective are first applied, to efficiently filter out negative examples, leaving the more computationally intensive tasks to the classifiers downstream in the dataflow network. A triaging scheme will also be developed in Phase II, in which a meta-level classifier learns which decision trees will most reliably classify each new SEC filing. We will investigate the use of triaging in the cascading paradigm as well, whereby an SEC filing is successively passed to whichever decision tree is the most appropriate, based on computational requirements and the classification results produced so far.

3.3. Management of SEC Filing Data

3.3.1. Numeric SEC Filing Data Extraction

SEC filing data was acquired by downloading 10-K and 10-Q Filings from Edgar and extracting the numeric data contained in the attached financial data schedules (FDSs). FDSs were generally attached to company filings dated between 1995 and 1999. The FDSs contain the original, unrevised data from the balance sheets and statements of income and cash flow. To support our numeric data analysis, twelve raw financial data features were extracted .

Downloading and extraction routines were implemented in Java, to acquire and store the numeric FDS data on a per-industry basis, as organized by SIC code, in XML-formatted files. FDSs were constructed manually for a small number of "important" filings which we had identified as red-flag examples, but lacked FDS attachments. FDS extraction was limited to 10-K filings in Phase I. We intend to widen our scope in Phase II to include both 10-K and 10-Q filings, in order to exploit additional evidence of creative accounting which is revealed in quarterly changes of financial metrics.

3.3.2. Derived Financial Metrics

Software was also developed for computing derived financial metrics and ratios required for red-flag analysis. The derived metrics include standard ratios, annual rates-of-change, and industry percentiles. The annual rate-of-change is computed for any raw financial metric or derived ratio, and industry percentiles are calculated for all static and rate-of-change features. Percentiles are organized by SIC (Standard Industrial Classification) code, supporting the detection of deviations from industry norms (e.g., to show that a company's annual increase in receivables is in the upper 95^{th} percentile for its SIC). Financial metric aberrations and deviations from industry norms are often associated with creative accounting practices.

3.3.3. Red-flag filing examples

Our semi-automatic approach to generating decision trees requires trainings sets of positive and negative class instances of "red-flag" and benign SEC filings. Positive class examples of

---

Zenko, B., Todorovski, L., Dzeroski, S. "A comparison of stacking with meta decision trees to other combining methods". *Proceedings A of the Fourth International Multi-Conference Information Society IS'2001,* pp. 144-147. Jozef Stefan Institute, Ljubljana, Slovenia, 2001.

creative accounting were obtained from SEC litigation releases related to inflated or fraudulent revenue recognition. Listed in Appendix B, 49 red-flag filings were identified which were identified in SEC litigation releases. Together, the 49 "positive class" examples span 19 SICs. For purposes of decision tree generation, "negative class" training sets were obtained as random samples of the nonimplicated filings submitted by companies belonging to the same SICs as the red-flag filings.

The "positive" creative accounting examples were obtained by downloading all of the SEC litigation releases available at the SEC website, from September 1995 through August 2002, and identifying those related to inflated or fraudulent revenue claims. Litigation releases that did not clearly state the filing period in question, or identified only quarterly filings, were ignored, since we have focused on 10-K filings for Phase I. Fraudulent filings identified before 1994 were also ignored, since their FDSs are generally unavailable. In some cases, the implicated 10-K filings were available but had no attached FDS, so we constructed the FDS manually from the filing content.

3.3.4. SEC Data Repository

We employed our distributed query engine *Genre* to manage the large quantities of SEC filing records available for decision tree analysis. Financial data schedules extracted from SEC filings are stored in XML format, supplemented with derived financial metrics, and ratios, and converted into a Genre repository. The datamining client downloads sets of SEC filing records from the Genre server, based upon user-specified constraints on the filing period, metric ranges, etc. To supplement the features available for decision rule formation, the raw and derived metrics are augmented with rate-of-change and SIC-percentile features. Summary data records supporting SIC percentile calculations are stored separately in the repository (by SIC, metric, and period) and downloaded to the client on a per-need basis.

4. Conclusion and future directions

We believe that the decision tree approach detects misleading information for numerical data. This approach needs to be extended in several directions to create a product at the end of Phase-II:

- **Scalability**: The system needs to scale in terms of number of filings that can be analyzed and types of filings without requiring hand coding of derived values. A framework needs to be developed to allow domain specific derivation calculations to be added to the system.
- **Learning**: The learning system needs to be extended to cover non-numeric domains.
- **Large volume and continuous input**: The system needs to scale tera-bytes of input data from different domains and to deal with cases where the filing is changing on a continuous basis.
- **Bayesian reasoning**: The probabilistic reasoning needs to be better integrated with the decision tree used for classification such that Bayesian reasoning techniques can be used to extend the learning power of the system.
- **Use of multiple classifiers**: The system needs to be extended and tested to deal with an ensemble of classifiers that may improve the quality of detection. Additionally, the system needs to be extended to use results produced by classifiers that work on textual data and link detection classifiers in order to deal with free-text and in order to deal with organizational fraud.

We believe that these extensions could be implemented in Phase-II. We have used appropriate frameworks developed from other DARPA projects to address volume scalability and classifier compositions. We can extend this framework to address the points presented above.

5. Appendix A: Financial Data Features

The financial data features used in Phase I are listed below. The "raw" data features are a subset of those included in the financial data schedules attached to the 10-K filings. The derived features include standard and ad-hoc ratios calculated from the raw features.

5.1. A.1. Raw features

ALLOWANCES
CASH
CGS
CURRENT-ASSETS
CURRENT-LIABILITIES
DEPRECIATION
INVENTORY
NET-INCOME
PP&E
RECEIVABLES
SALES
TOTAL-ASSETS
TOTAL-LIABILITY-AND-EQUITY
TOTAL-REVENUES

5.2. A.2. Derived features

Derived Feature	Formula
CURRENT-RATIO	CURRENT-ASSETS / CURRENT-LIABILITIES
CGS-TVR-RATIO	TOTAL-REVENUES / CGS
DAYS-INVENTORY-OUTSTANDING	365 * INVENTORY / CGS
DAYS-SALES-OUTSTANDING	365 * RECEIVABLES / TOTAL-REVENUES
INVENTORY-TVR-RATIO	TOTAL-REVENUES / INVENTORY
PP&E-TVR-RATIO	TOTAL-REVENUES / PP&E
QUICK-RATIO	(CURRENT-ASSETS - INVENTORY) / CURRENT-LIABILITIES
RECEIVABLES-TVR-RATIO	TOTAL-REVENUES / RECEIVABLES
REVENUE-CGS-RATIO	TOTAL-REVENUES / CGS
TOTAL-ASSETS-TVR-RATIO	NET-INCOME / TOTAL-ASSETS

6. Appendix B: Red-flag 10-K filing examples

No.	SIC	Company	Fiscal Year	Litigation Release
1	2300	Boss Holdings	1995	LR-17044.
2	2300	Boss Holdings	1996	LR-17044
3	2390	Sunbeam	1996	LR-17001
4	2390	Sunbeam	1997	LR-17001
5	2834	Amazon Natural Treasures	1998	LR-16924
6	2840	USA Detergents	1996	LR-17426
7	3460	Advanced Technical Products	1998	LR-17074
8	3480	Madera International	1998	LR-17140
9	3577	Centennial Technologies	1996	LR-16725
10	3577	Cylink	1997	LR-16728
11	3577	Xerox	1997	LR-17465
12	3577	Xerox	1998	LR-17465
13	3577	Xerox	1999	LR-17465
14	3661	PictureTel	1996	LR-17448
15	3663	EFJ	1996	LR-16887
16	3669	Vari L Company	1996	LR-17671
17	3669	Vari L Company	1997	LR-17671
18	3669	Vari L Company	1998	LR-17671
19	3669	Vari L Company	1999	LR-17671
20	3690	Aura Systems	1997	LR-17155
21	3690	Aura Systems	1998	LR-17155
22	3690	NewCom	1998	LR-17557
23	3825	Signal Technology	1996	LR-17439
24	3825	Signal Technology	1997	LR-17439
25	4953	Waste Management	1996	LR-17435
26	4953	Waste Management	1997	LR-17435
27	4955	Solucorp	1997	LR-16785
28	4955	Solucorp	1998	LR-16785
29	5063	Anicom	1998	LR-17504
30	5063	Anicom	1999	LR-17504
31	6162	AppOnline	1997	LR-17407
32	6162	AppOnline	1998	LR-17407
33	6162	AppOnline	1999	LR-17407
34	7370	Physician Computer Network	1996	LR-17542
35	7372	Accelr8	1998	LR-16354
36	7372	AremisSoft	2000	LR-17172
37	7372	Informix	1996	LR-16757
38	7372	Legato Systems	1999	LR-17524
39	7372	MicroStrategy	1998	LR-16829
40	7372	MicroStrategy	1999	LR-16829
41	7372	System Software Associates	1996	LR-16627
42	7372	Unify	2000	LR-17522
43	7372	Versatility	1997	LR-16709
44	7389	American Bank Note Holographics	1998	LR-17068
45	7389	Critical Path	2000	LR-17353
46	7389	Itex	1996	LR-16536
47	7389	Itex	1997	LR-16536
48	7822	Livent	1995	LR-16022
49	7822	Livent	1996	LR-16022